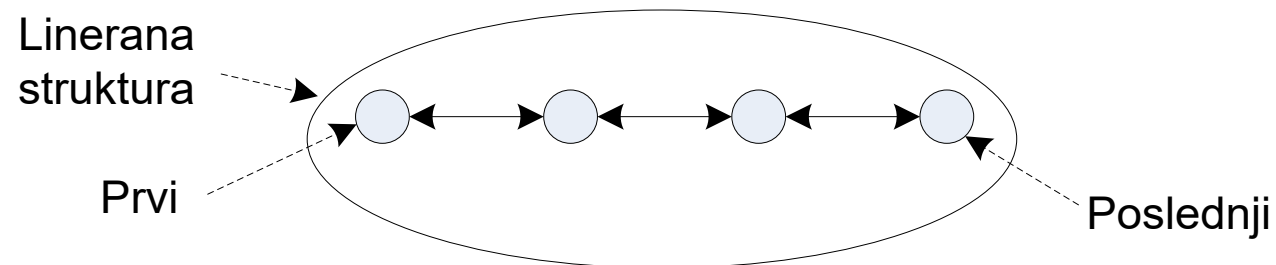


Jednostruko povezana lista

Linearne strukture podataka

- Svakom element linearne strukture osim prvog prethodi tačno jedan element
- Svakom element linearne strukture osim poslednjeg sledi tačno jedan element
- Primeri linearnih struktura:
 - Liste
 - Stekovi
 - Redovi



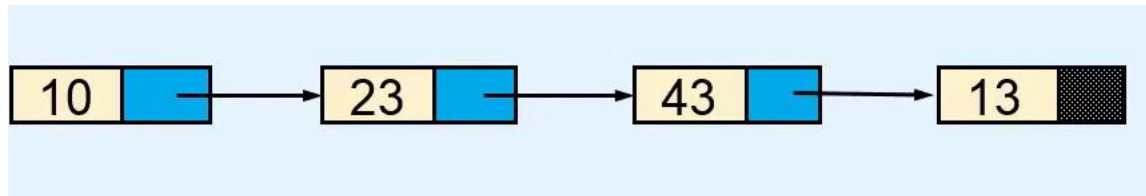
Liste

- Dinamička struktura podataka sačinjena od elemenata koji se nazivaju čvorovi
- Podaci se ne čuvaju u uzastopnim memorijskim lokacijama
- Veličina niza se unapred definiše dok veličina liste dinamički raste
- Unos i brisanje elemenata su manje kompleksne operacije nego kod nizova
- Nije moguć direktni pristup elementu liste, već se elementu pristupa sekvencijalno
- Liste zahtevaju dodatni prostor za čuvanje pointera na sledeći član liste

Jednostruko povezana lista

- Jednostruko povezana lista je skup čvorova povezanih pokazivačima u jednom smeru
- Svaki element liste osim poslednjeg ima jednog sledbenika
- Krajnji čvor liste ne pokazuje na neki čvor, odnosno sadrži NULL pokazivač
- Čvor liste se sastoji iz dva dela:
 - informacionog dela, gde se nalaze podaci
 - link deo koji pokazuje na sledeći čvor

Primer jednostruko povezane liste



Klasa Node

```
public class Node
{
    public int info;
    public Node link;

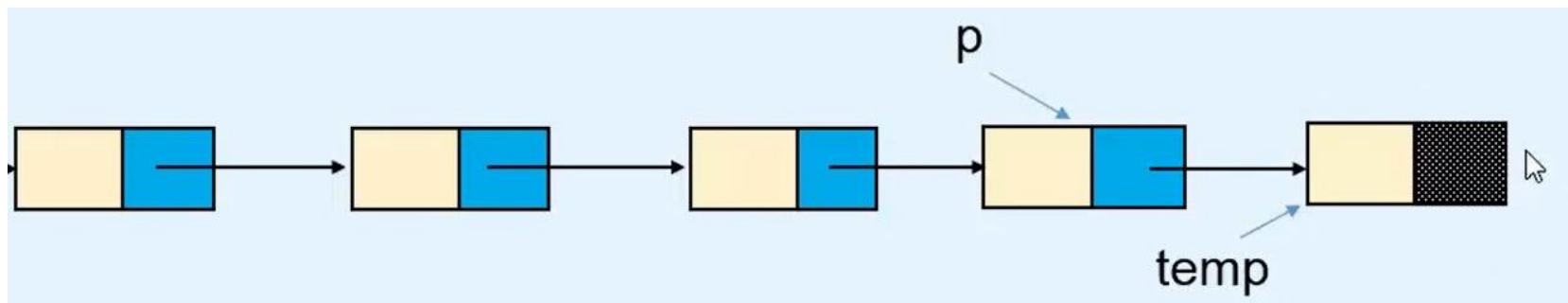
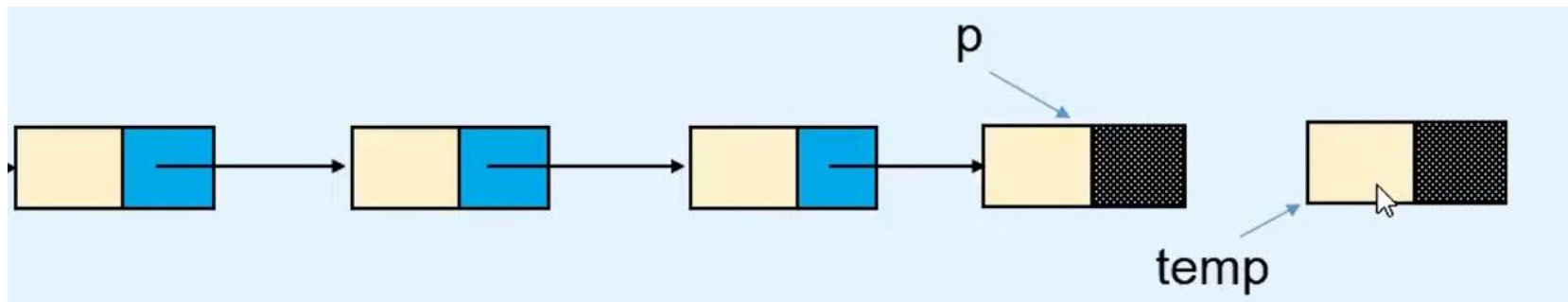
    public Node(int info)
    {
        this.info = info;
        link = null;
    }
}
```

Klasa LinkedList

```
public class LinkedList
{
    private Node start;

    public LinkedList()
    {
        start= null;
    }
}
```

Dodavanje čvora na kraj liste



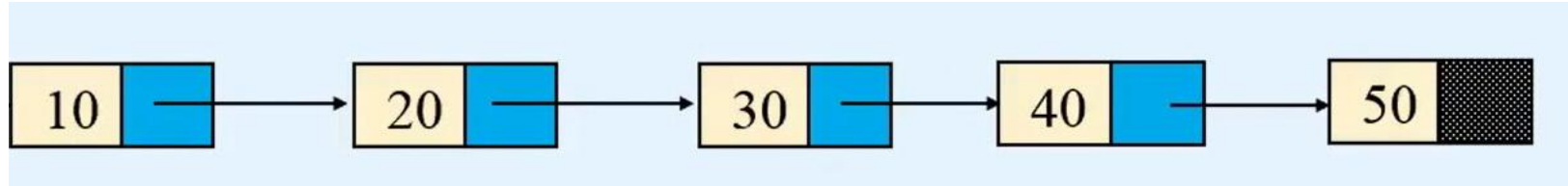
Dodavanje čvora na kraj liste

```
public void DodajNaKraj(int info)
{
    //kreiranje novog objekta klase Node
    Node temp = new Node(info);

    //ako je lista prazna
    if (start == null)
    {
        start = temp; // kreirani cvor je prvi element liste
    }
    else
    {
        Node p = start;

        while (p.link != null)
        {
            p = p.link;
        }
        p.link = temp;
    }
}
```

Prolazak kroz listu



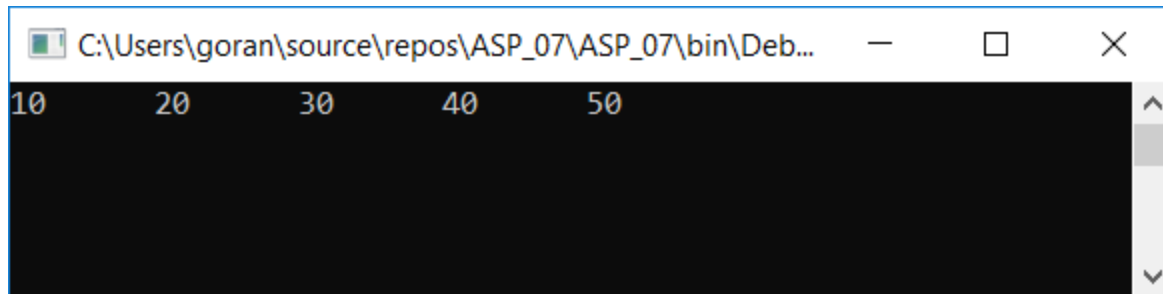
```
public void PrikaziListu()
{
    Node p = start;

    if (p == null)
    {
        Console.WriteLine("Lista je prazna");
    }
    while (p != null)
    {
        Console.Write(p.info + "\t");
        p = p.link;
    }
    Console.WriteLine();
}
```

Kreiranje liste

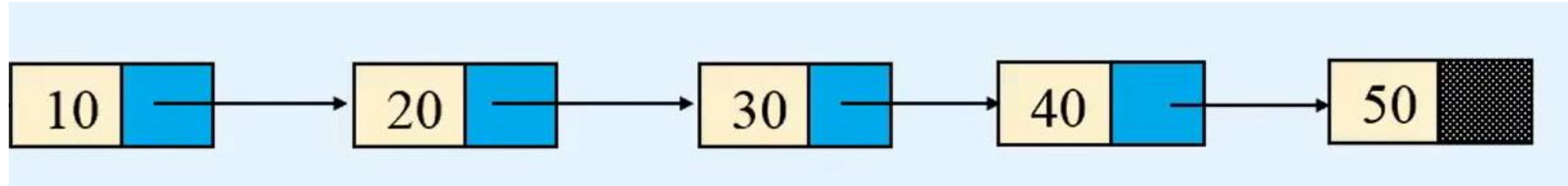
```
static void Main(string[] args)
{
    LinkedList lista1 = new LinkedList();

    lista1.DodajNaKraj(10);
    lista1.DodajNaKraj(20);
    lista1.DodajNaKraj(30);
    lista1.DodajNaKraj(40);
    lista1.DodajNaKraj(50);
    lista1.PrikaziListu();
    Console.ReadLine();
}
```

A screenshot of a Windows console application window. The title bar shows the file path "C:\Users\goran\source\repos\ASP_07\ASP_07\bin\Deb...". The console output displays the numbers 10, 20, 30, 40, and 50 on a single line, separated by spaces. The rest of the console area is black, indicating that the program has finished execution and is waiting for input.

```
C:\Users\goran\source\repos\ASP_07\ASP_07\bin\Deb...
10 20 30 40 50
```

Brojanje čvorova



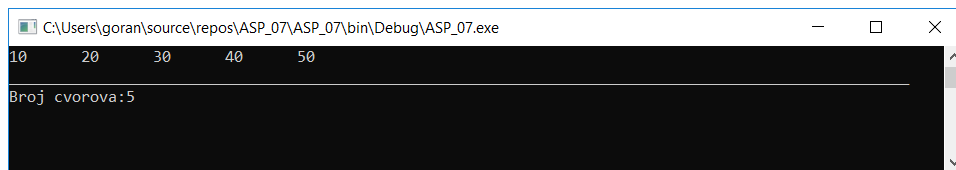
```
public int BrojCvorova()
{
    int n = 0;
    Node p = start;

    while (p != null)
    {
        n++;
        p = p.link;
    }
    return n;
}
```

Brojanje čvorova

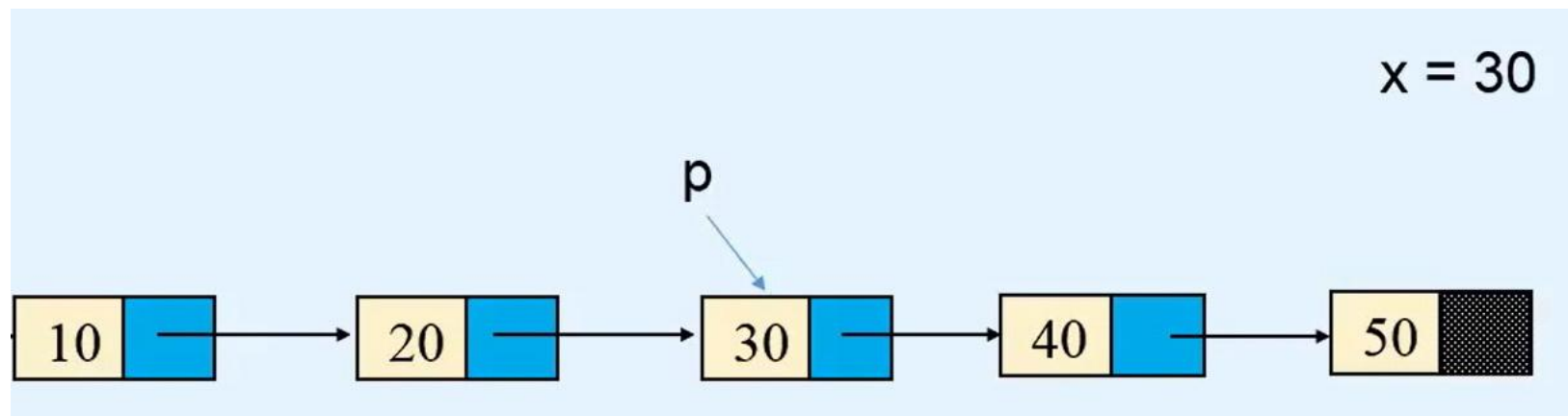
```
static void Main(string[] args)
{
    LinkedList lista1 = new LinkedList();

    lista1.DodajNaKraj(10);
    lista1.DodajNaKraj(20);
    lista1.DodajNaKraj(30);
    lista1.DodajNaKraj(40);
    lista1.DodajNaKraj(50);
    lista1.PrikaziListu();
    Linija(100);
    Console.WriteLine("Broj cvorova:" + lista1.BrojCvorova());
    Console.ReadLine();
}
```



The screenshot shows a Windows console window titled "C:\Users\goran\source\repos\ASP_07\ASP_07\bin\Debug\ASP_07.exe". The console output displays a horizontal line with five numbers: 10, 20, 30, 40, and 50. Below this line, the text "Broj cvorova:5" is printed, indicating the total number of nodes in the linked list.

Pronalaženje elementa liste na osnovu vrednosti



Pronalaženje elementa liste na osnovu vrednosti

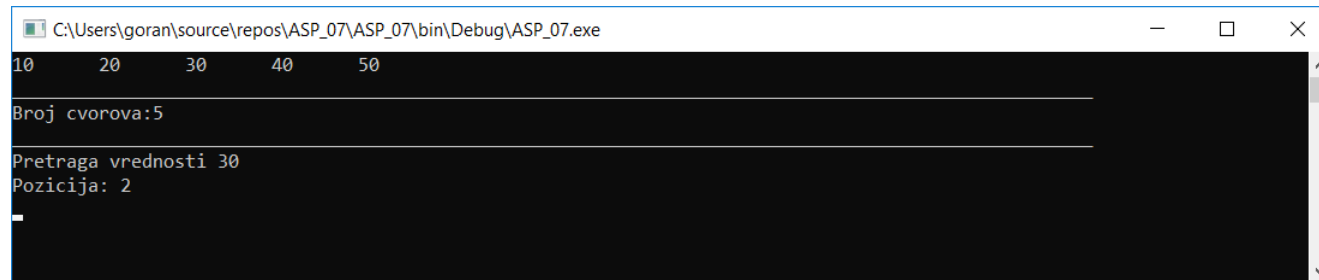
```
public int PronadjiVrednost(int x)
{
    Node p = start;
    int pozicija = 0;

    while (p != null)
    {
        if (p.info == x)
        {
            break;
        }
        p = p.link;
        pozicija++;
    }

    if (p == null)
    {
        return -1;
    }
    else
    {
        return pozicija;
    }
}
```

Pretraga vrednosti

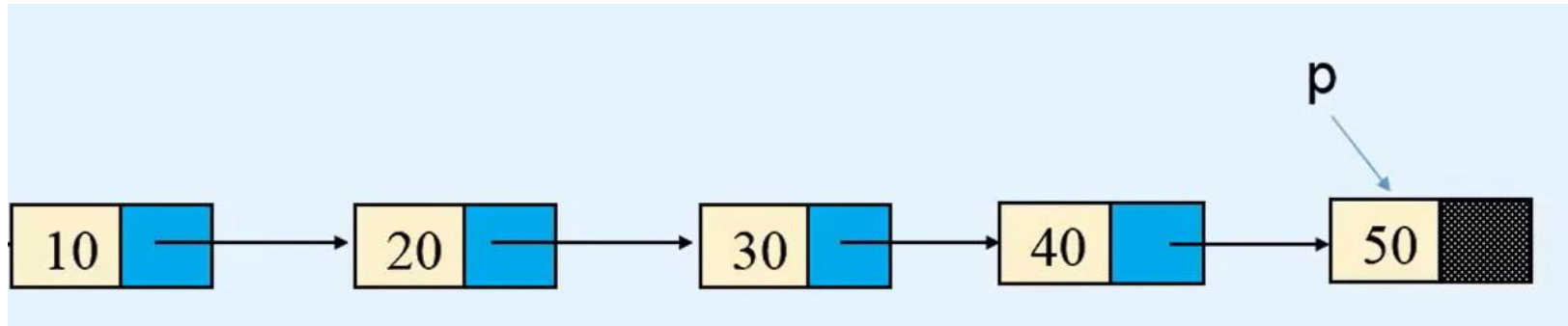
```
//Pronadji vrednost  
Console.WriteLine("Pretraga vrednosti 30");  
int ind= lista1.PronadjiVrednost(30);  
Console.WriteLine($"Pozicija: {ind}");
```



The screenshot shows a console window titled "C:\Users\goran\source\repos\ASP_07\ASP_07\bin\Debug\ASP_07.exe". The output of the program is as follows:

```
10    20    30    40    50  
_____  
Broj cvorova:5  
_____  
Pretraga vrednosti 30  
Pozicija: 2  
_____
```

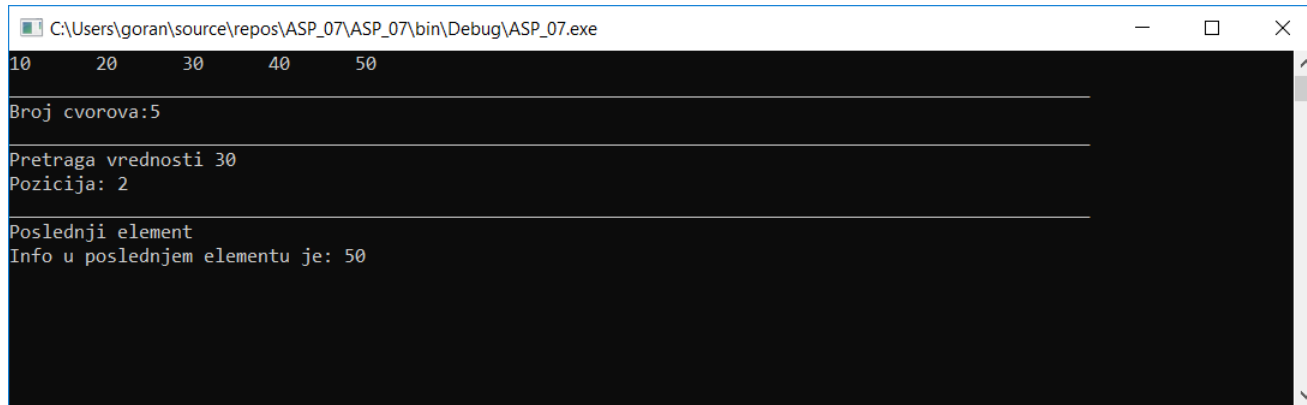
Referenca na poslednji čvor liste



```
public Node PronadjiPoslednji()
{
    Node p = start;
    while (p.link != null)
    {
        p = p.link;
    }
    return p;
}
```

Prikaz poslednjeg elementa liste

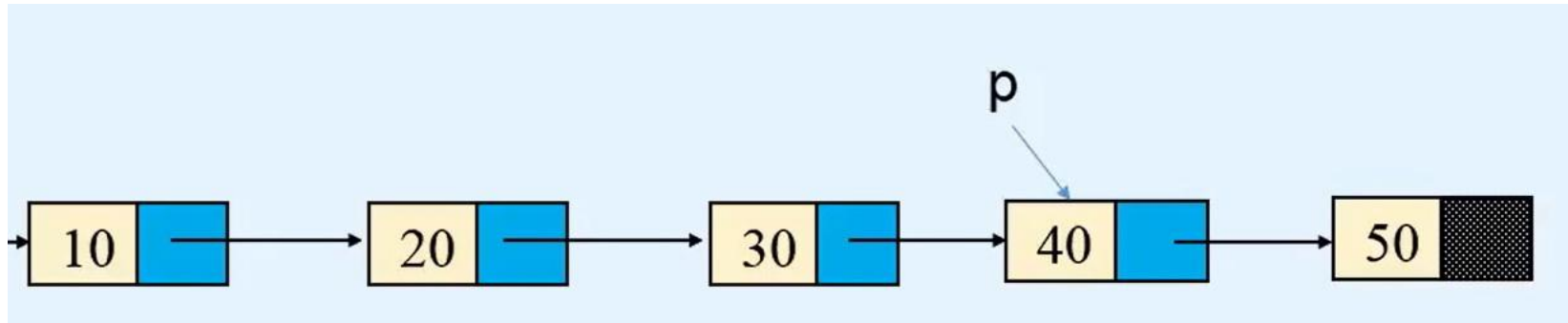
```
Console.WriteLine("Poslednji element");  
Node p = lista1.PronadjiPoslednji();  
Console.WriteLine("Info u poslednjem elementu je: " + p.info);
```



A screenshot of a Windows console application window. The title bar shows the file path: C:\Users\goran\source\repos\ASP_07\ASP_07\bin\Debug\ASP_07.exe. The console output is as follows:

```
10      20      30      40      50  
Broj cvorova:5  
Pretraga vrednosti 30  
Pozicija: 2  
Poslednji element  
Info u poslednjem elementu je: 50
```

Referenca na predposlednji čvor liste



```
public Node PronadjiPredPoslednji()
{
    Node p = start;
    while (p.link.link != null)
    {
        p = p.link;
    }
    return p;
}
```

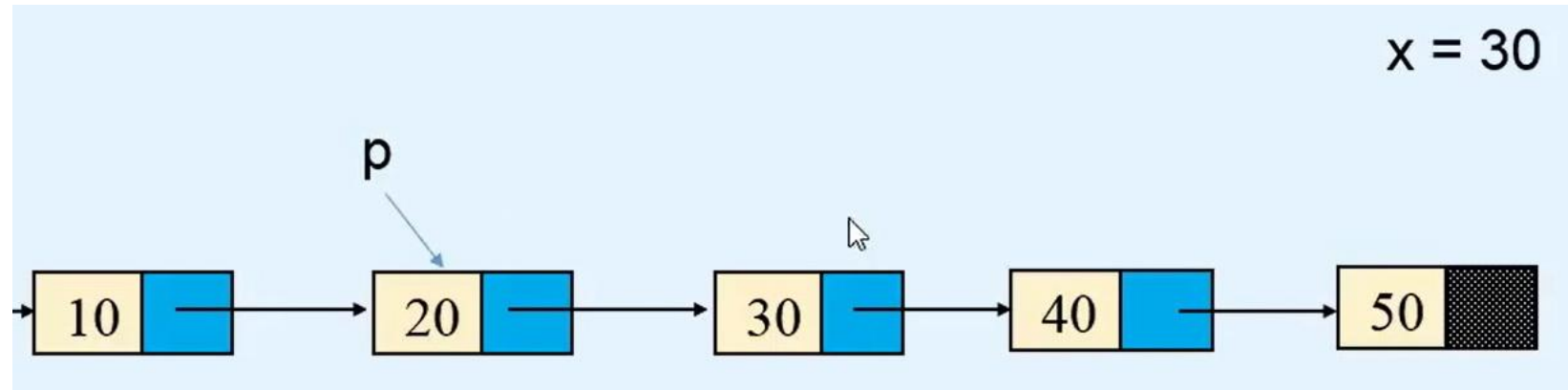
Prikaz predposlednjeg elementa liste

```
p = lista1.PronadjiPredPoslednji();  
Console.WriteLine("Info u predposlednjem  
elementu je: " + p.info);
```



```
C:\Users\goran\source\repos\ASP_07\ASP_07\bin\Debug\ASP_07.exe  
10 20 30 40 50  
Broj cvorova:5  
Pretraga vrednosti 30  
Pozicija: 2  
Poslednji element  
Info u poslednjem elementu je: 50  
Info u predposlednjem elementu je: 40
```

Pronalaženje prethodnog elementa



Pronalaženje elementa čiji sledeći element ima vrednost $x=30$

Pronalaženje prethodnog elementa

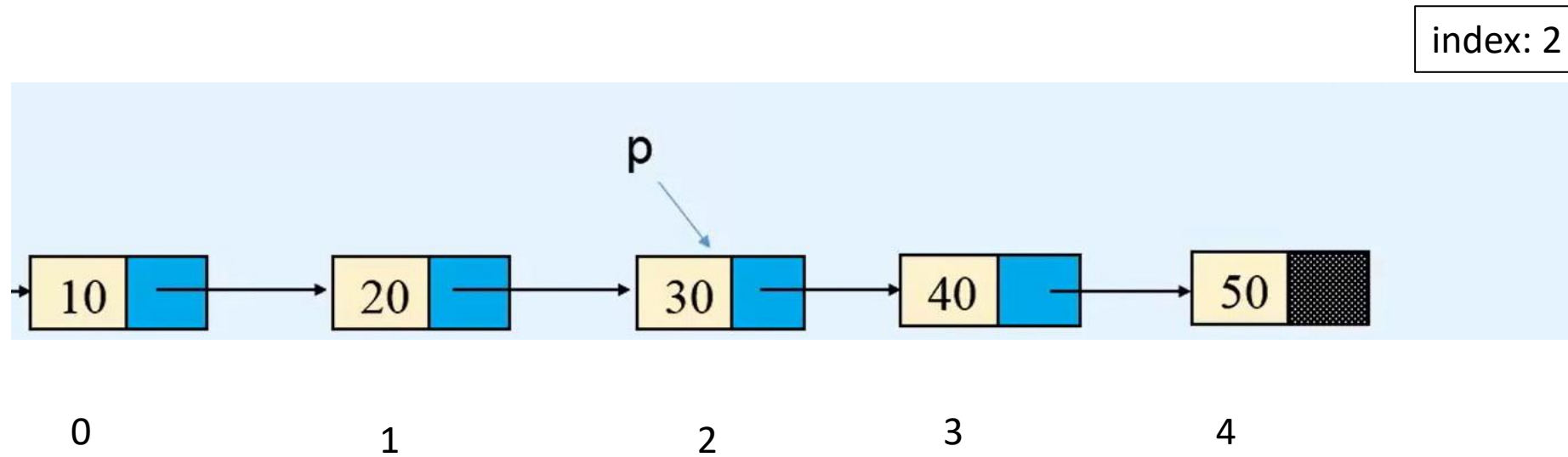
```
public Node PronadjiPrethodni(int x)
{
    Node p = start;
    while (p.link != null)
    {
        if (p.link.info == x)
        {
            break;
        }
        p = p.link;
    }
    if (p.link == null)
    {
        return null;
    }
    return p;
}
```

Pronalaženje prethodnog elementa

```
Console.WriteLine("Pronadji prethodni element za element koji sadrzi vrednost 30");  
p = lista1.PronadjiPrethodni(30);  
Console.WriteLine("Info u prethodnom elementu je: " + p.info);
```

```
C:\Users\goran\source\repos\ASP_07\ASP_07\bin\Debug\ASP_07.exe  
10 20 30 40 50  
Broj cvorova:5  
Pretraga vrednosti 30  
Pozicija: 2  
Poslednji element  
Info u poslednjem elementu je: 50  
Info u predposlednjem elementu je: 40  
Pronadji prethodni element za element koji sadrzi vrednost 30  
Info u prethodnom elementu je: 20
```

Pronalaženje elementa na poziciji

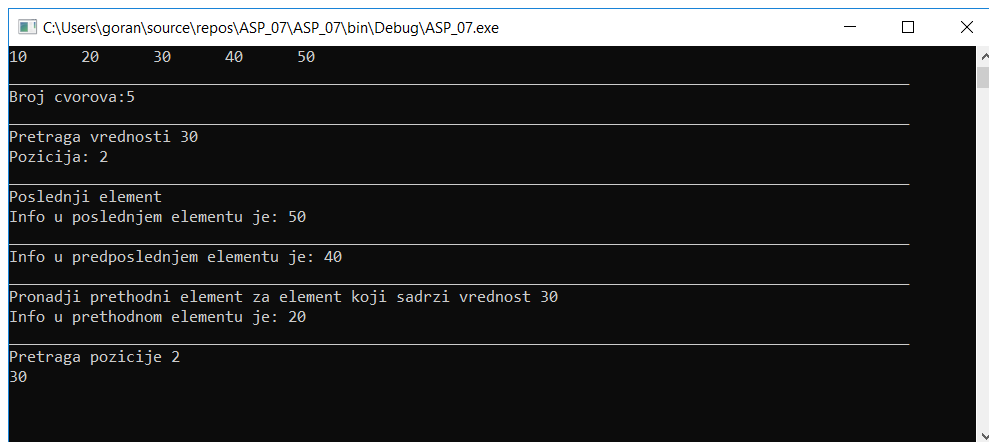


Pronalaženje elementa na poziciji

```
public Node Pronadji(int index)
{
    Node p = start;
    for (int i = 0; i < index && p != null; i++)
    {
        p = p.link;
    }
    return p;
}
```

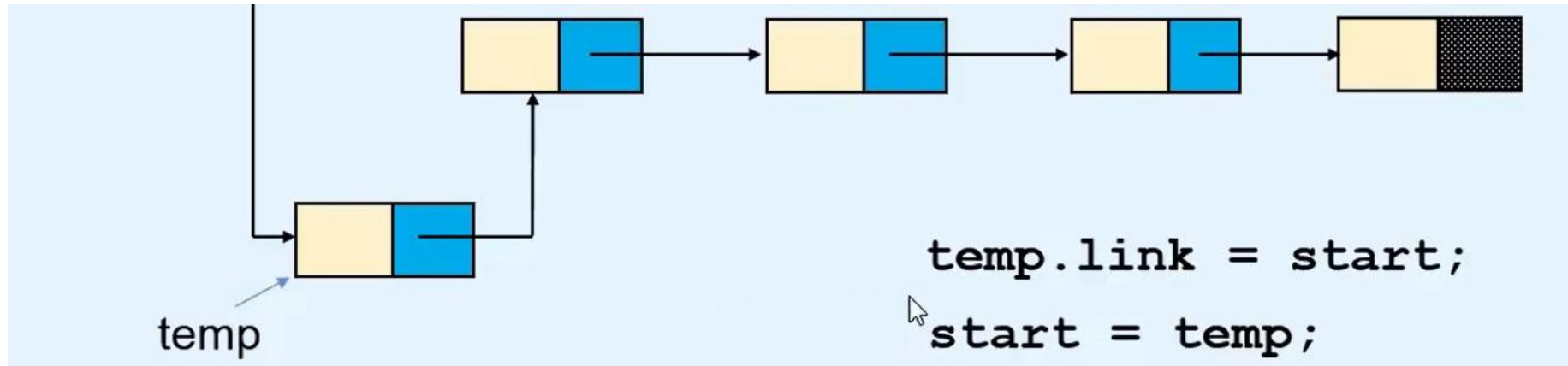
Pronalaženje elementa na poziciji

```
Console.WriteLine("Pretraga pozicije 2");  
p = lista1.Pronadji(2);  
if (p != null)  
{  
    Console.WriteLine(p.info);  
}  
else  
{  
    Console.WriteLine("Ne postoji");  
}
```



```
C:\Users\goran\source\repos\ASP_07\ASP_07\bin\Debug\ASP_07.exe  
10    20    30    40    50  
Broj cvorova:5  
Pretraga vrednosti 30  
Pozicija: 2  
Poslednji element  
Info u poslednjem elementu je: 50  
Info u predposlednjem elementu je: 40  
Pronadji prethodni element za element koji sadrzi vrednost 30  
Info u prethodnom elementu je: 20  
Pretraga pozicije 2  
30
```

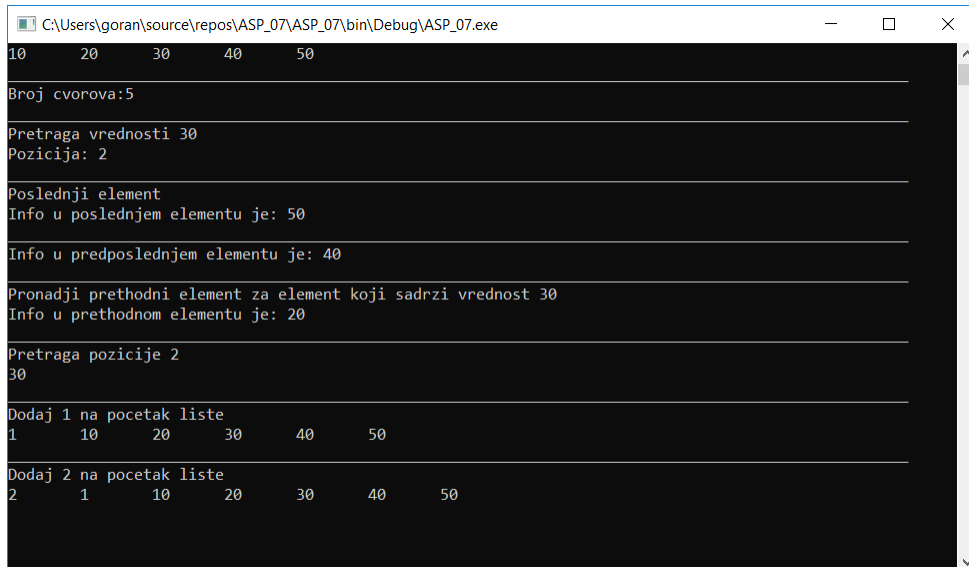
Ubacivanje novog čvora na početak liste



```
public void DodajNaPocetak(int info)
{
    Node temp = new Node(info);
    temp.link = start;
    start = temp; // novi cvor postaje pocetni element
}
```

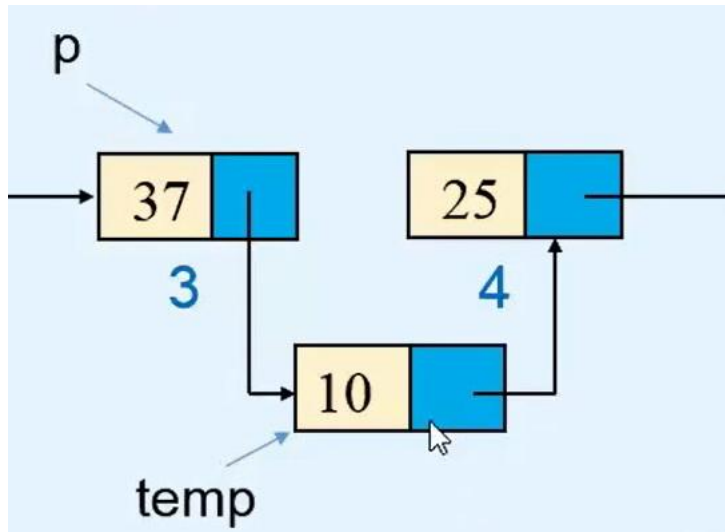
Dodavanje na početak liste

```
Console.WriteLine("Dodaj 1 na pocetak liste");  
lista1.DodajNaPocetak(1);  
lista1.PrikaziListu();  
Linija(100);  
Console.WriteLine("Dodaj 2 na pocetak liste");  
lista1.DodajNaPocetak(2);  
lista1.PrikaziListu();
```



```
C:\Users\goran\source\repos\ASP_07\ASP_07\bin\Debug\ASP_07.exe  
10    20    30    40    50  
Broj cvorova:5  
Pretraga vrednosti 30  
Pozicija: 2  
Poslednji element  
Info u poslednjem elementu je: 50  
Info u predposlednjem elementu je: 40  
Pronadji prethodni element za element koji sadrzi vrednost 30  
Info u prethodnom elementu je: 20  
Pretraga pozicije 2  
30  
Dodaj 1 na pocetak liste  
1      10    20    30    40    50  
Dodaj 2 na pocetak liste  
2      1     10    20    30    40    50
```

Ubacivanje čvora na poziciju



```
temp.link = p.link  
p.link = temp
```

- Ubacujem čvor na poziciju **index=4**
- Ubačeni čvor pokazuje na čvor na koga je pokazivao čvor na poziciji **index-1=3**
- Čvor koji je na poziciji **index-1 = 3** pokazuje na čvor koji ubacujemo

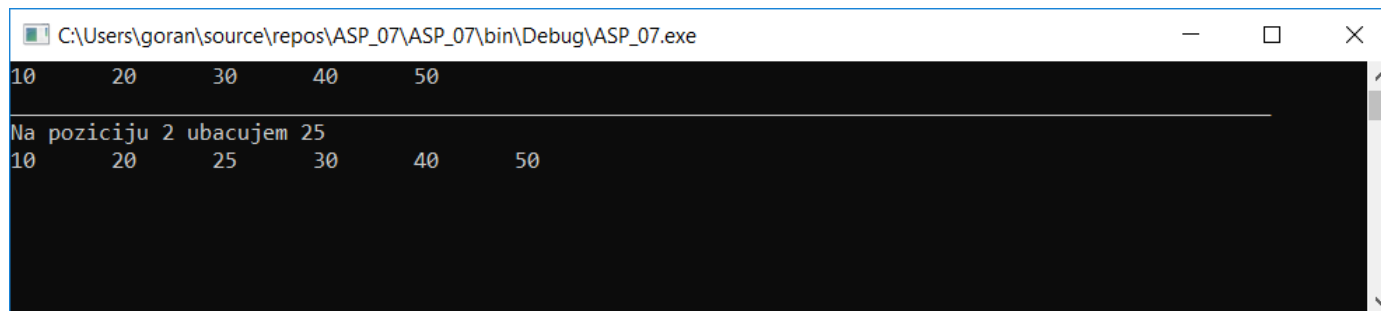
Ubacivanje čvora na poziciju - kod

```
public void DodajNaPoziciju(int index, int info)
{
    Node temp = new Node(info);

    if (index == 0)
    {
        DodajNaPocetak(info);
    }
    else
    {
        Node p = start;
        for (int i = 0; i < index - 1; i++)
        {
            p = p.link;
        }
        //p je sada cvor na poziciji index-1
        temp.link = p.link;
        p.link = temp;
    }
}
```

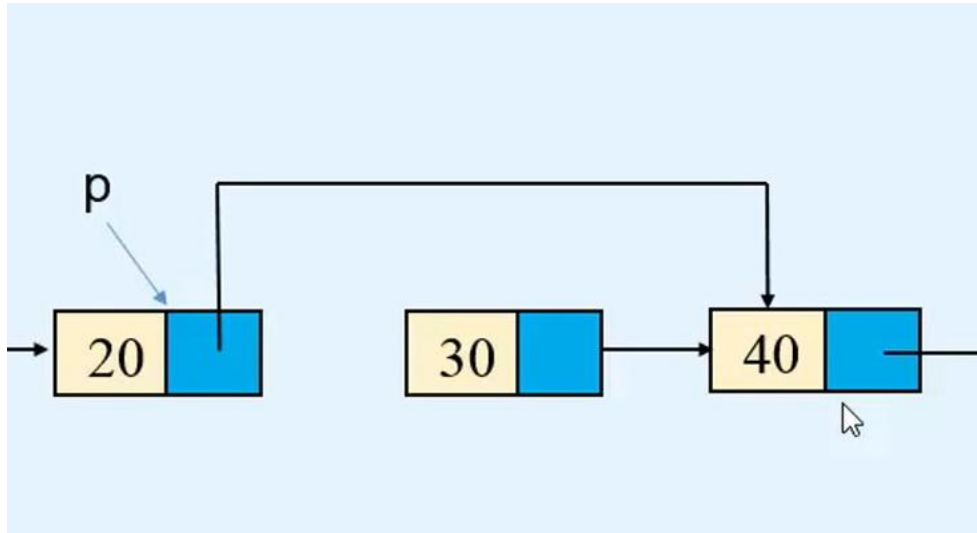
Ubacivanje čvora na poziciju

```
LinkedList lista1 = new LinkedList();  
  
lista1.DodajNaKraj(10);  
lista1.DodajNaKraj(20);  
lista1.DodajNaKraj(30);  
lista1.DodajNaKraj(40);  
lista1.DodajNaKraj(50);  
lista1.PrikaziListu();  
Linija(100);  
Console.WriteLine("Na poziciju 2 ubacujem 25");  
lista1.DodajNaPoziciju(2, 25);  
lista1.PrikaziListu();
```



```
C:\Users\goran\source\repos\ASP_07\ASP_07\bin\Debug\ASP_07.exe  
10    20    30    40    50  
Na poziciju 2 ubacujem 25  
10    20    25    30    40    50
```

Uklanjanje elementa iz liste



- p je prethodni element elementa koga brišemo
- p.link.link je sledeći element koji dolazi iza elementa koga brišemo
- p.link = p.link.link – uklanja element koji dolazi posle elementa p

Brisanje elementa sa pozicije

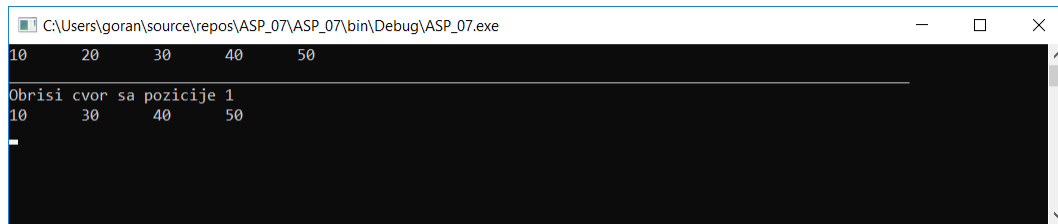
```
public void ObrisiCvor(int index)
{
    //prazna lista
    if (start == null)
    {
        return;
    }
    // brise se pocetak liste
    if (index == 0)
    {
        start = start.link;
        return;
    }
    Node p = start;
    // Nadji prethodni cvor cvora koga brisemo
    for (int i = 0; i < index - 1; i++)
    {
        p = p.link;
    }

    // Nadji element koji sledi iza elementa koga brisemo
    Node sledeci = p.link.link;

    p.link = sledeci;
}
```

Brisanje elementa sa pozicije

```
lista1.DodajNaKraj(10);  
lista1.DodajNaKraj(20);  
lista1.DodajNaKraj(30);  
lista1.DodajNaKraj(40);  
lista1.DodajNaKraj(50);  
lista1.PrikaziListu();  
Linija(100);  
Console.WriteLine("Obrisi cvor sa pozicije 1");  
lista1.ObrisiCvor(1);  
lista1.PrikaziListu();
```



```
C:\Users\goran\source\repos\ASP_07\ASP_07\bin\Debug\ASP_07.exe  
10    20    30    40    50  
Obrisi cvor sa pozicije 1  
10    30    40    50
```

Pitanje 1

Dat je sledeći kod:

```
Node p = start;  
while (p.link != null)  
{  
    p = p.link;  
}
```

Nakon izvršavanja koda p će biti:

- a. Referenca na poslednji čvor liste
- b. Referenca na prvi čvor liste
- c. Null referenca

Odgovor: a

Pitanje 2

Unos novog čvora temp na početak liste čije je prvi element start vrši se sledećim linijama koda:

- a. `start = temp;`
 `temp.link = start;`
- b. `temp.link = start;`
 `start = temp;`
- c. `temp.link = start;`
 `temp= start;`

Odgovor: b

Pitanje 3

Brisanje čvora koji dolazi posle čvora p iz liste vrši se sledećim kodom:

- a. `p.link=null;`
- b. `p.link=p;`
- c. `p.link = p.link.link;`

Odgovor: c

Pitanje 4

Čvorovi jednostruko povezane liste čuvaju se u uzastopnim memorijskim lokacijama:

- a. Da
- b. Ne

Odgovor: b